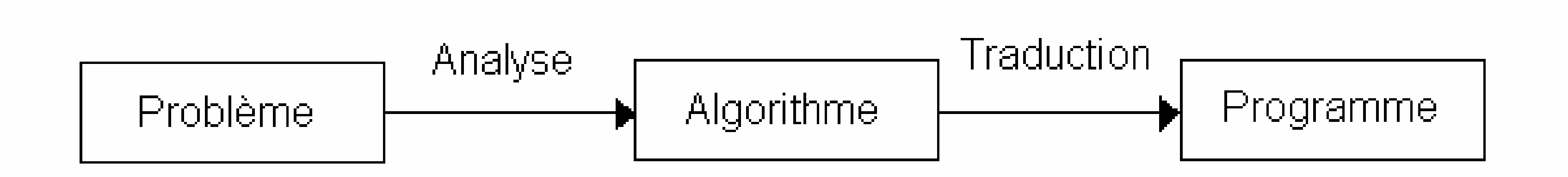
Chapitre 2
Algorithmique

2.1 Notion d'algorithme et de programme

- Définition : un ordinateur est une machine électronique, qui permet de traiter des informations selon des séquences d'instructions prédéfinies (un **programme**).

- **Notre objectif => Apprendre à écrire des programmes.**



2.1 Notion d'algorithme et de programme

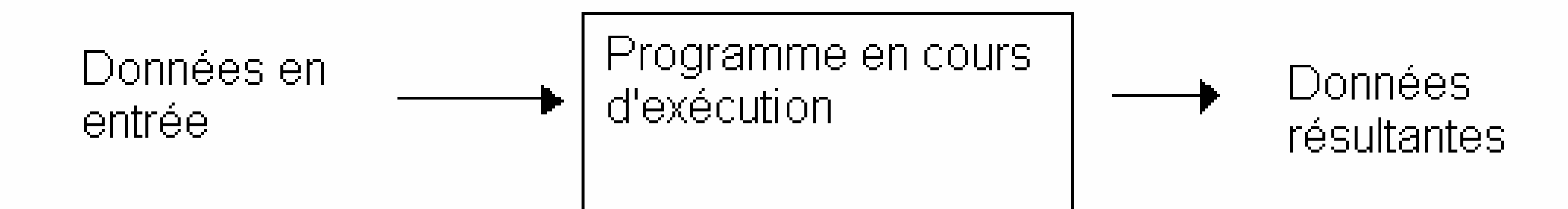
- Problèmes :
  - Addition de deux nombres
  - Calcul de la factorielle d'un nombre
  - Dessiner la courbe d'une fonction
  - Calculer la moyenne des notes des étudiants

- Algorithme : une suite organisée d'opérations élémentaires pour l'obtention d'une solution à un problème posé.

- Programme : une séquence d'instructions écrites dans un langage de programmation indiquant à un ordinateur ce qu'il doit faire.

2.1 Notion d’algorithme et de programme

- Langage de programmation : c’est le langage dans lequel on écrit un programme informatique. Exemple : **Fortran**, Pascal, C++, Java, …
- Exécution d’un programme :
  - L’exécution se fait en séquence : exécuter la première instruction, puis la deuxième puis la troisième, …
  - Pendant l’exécution :



2.2 Analyse d'un problème

- Les étapes d'analyse :
  - Comprendre le problème.
  - Dégager les données en entrée.
  - Dégager les données en sortie.
  - Formaliser la solution trouvée par un algorithme.
- Exemple d'algorithme : Addition de deux nombres A et B.
  - Introduire le premier nombre A
  - Introduire le deuxième nombre B
  - Faire l'addition A+B et mettre le résultat dans C
  - Afficher C
- Une opération dans un algorithme est équivalente à une action sur un objet. Dans l'algorithme précédent :
  - Actions : Introduire, faire l'addition, mettre le résultat dans, Afficher.
  - Objets : les nombres A, B, C.

2.3 Les objets dans un algorithme

- Un objet possède trois caractéristiques : le nom, le type et la valeur.
  - Le nom (identificateur) de l'objet sert à le désigner dans l'algorithme. Il peut être alphanumérique et doit commencer par une lettre.
  - Le type de l'objet caractérise l'ensemble des valeurs permises pour cet objet et les opérations qui lui sont autorisées.
  - La valeur est le contenu de l'objet. Cette valeur peut changer pendant l'exécution de l'algorithme (si l'objet est une variable) ou non (si l'objet est une constante).
- Les constantes : une constante est un objet qui a un nom fixe, un type fixe, et une valeur fixe.
- Les variables : une variable est un objet qui a un nom fixe, un type fixe, et une valeur variable.

2.3 Les objets dans un algorithme

- Une variable peut être vue comme une place dans la mémoire centrale de l'ordinateur portant un nom.

	Nom	Type	Valeur
Constante	fixe	fixe	fixe
Variable	fixe	fixe	variable

M.C

V

2.4 Formalisme d'un algorithme

- Un algorithme est exprimé dans un langage proche du langage naturel et constitué d’une série de conventions, appelé **pseudo-code** (ou langage algorithmique). Ce langage permet de décrire de manière complète et claire les objets manipulés par l’algorithme ainsi que l’ensemble des actions à exécuter sur ces objets.
- Le format d’un algorithme est comme suit :

  <u>Algorithme</u> <nom-de-l’algorithme>

  <Déclaration des constantes est des variables>

  <u>Début</u>

  <Action 1>

  …

  <u>Fin</u>

2.4 Formalisme d'un algorithme

- **Déclaration des constantes :**

  Constante

  <Nom-de-constante1> = <valeur1>

  <Nom-de-constante2> = <valeur2>

- **Déclaration des variables :**

  Variable

  <Nom-de-variable1> : <Type1>

  <Nom-de-variable2> : <Type2>

- **Exemple :**

  Algorithme addition

  Constante

  pi = 3.14

  Variable

  A : réel

  B5 : entier

2.5 Les actions élémentaires

- **L'affectation:**

  C'est l'action par laquelle nous pouvons **attribuer** à une variable V une valeur résultante de l'évaluation d'une expression E. Le type de l'expression doit être compatible avec le type de la variable V. Cette action est notée par :

  V ← E

  Ce qui signifie évaluer E et ranger le résultat dans la zone mémoire appelée V.

- **Exemple:** la troisième action de l'algorithme d'addition

  C ← A+B

2.5 Les actions élémentaires

- **La lecture:**

  C’est l’action par laquelle nous pouvons introduire des données en utilisant le clavier. Elle est notée par :

  Lire (x)

  Ce qui signifie mettre dans la zone mémoire x la donnée tapée sur le clavier.

- **L’écriture:**

  C’est l’action par laquelle nous pouvons communiquer un résultat ou un message à l’utilisateur par l’intermédiaire de l’écran. Elle est notée par: **Écrire (E)**

  Ce qui signifie évaluer l’expression E et afficher le résultat.

2.5 Les actions élémentaires

- Exemple: addition de deux nombres entiers.

  Algorithme addition
  Variable
  A, B, C : entier
  Début
  Lire (A)
  Lire (B)
  C ← A + B
  Écrire (C)
  Fin

Exercice :

Donnez un algorithme qui permet de mettre les valeurs 5 et 19 dans deux variables entières A et B, puis d'échanger leurs contenus.

Solution:

Algorithme échange
Variable
A, B, C : entier
Début
A ← 5
B ← 19
C ← A
A ← B
B ← C
Fin

2.6 Les types élémentaires

- Les types élémentaires sont des types simples à valeur unique. On distingue les types : entier, réel, booléen, caractère et chaîne. Un type désigne l'ensemble des **valeurs** permises pour un objet, et l'ensemble des **opérateurs** autorisés sur les objets de ce type.
- **Le type entier :**
  - Désigne les nombres entiers. (ex: 2, +2, -30)
- **Opérateurs du type entier :**
  - Les opérateurs arithmétiques : +, -, *, /
  - L'opérateur modulo noté : mod (qui veut dire reste de la division)
  - Les opérateurs de relation : <, <=, >, >=, =, ≠

2.6 Les types élémentaires

- **Le type réel :**
  - Désigne les nombres réels. (ex: 2.5, +12.55, 2E5)
- **Opérateurs du type réel :**
  - Les opérateurs arithmétiques : +, -, *, /
  - Les opérateurs de relation : <, <=, >, >=, =, ≠
- **Le type booléen (logique):**
  - Une variable de ce type peut contenir une des valeurs logiques *vrai* ou *faux*.
- **Opérateurs du type booléen:**
  - Les opérateur logique: et, ou, non.
- **Exemple:**

  Constante

  V = vrai

  Variable

  test : booléen

2.6 Les types élémentaires

- **Le type caractère:**
  - Désigne l'ensemble des lettres alphabétiques majuscules et minuscules, les chiffres, les caractères spéciaux (ex: '?', '!', ')', '*', …) et le caractère espace.
- **Opérateurs du type caractère:**
  - Les opérateurs de relation : <, <=, >, >=, =, ≠, car les caractères sont ordonnés dans la table du code ASCII.
  - Les opérateurs *succ* et *pred*, qui signifient successeur et prédécesseur d'un caractère dans la table du code ASCII.
- Exemple:
  - succ('A') = 'B', pred ('C') = 'B'.

2.6 Les types élémentaires

- **Le type chaîne:**
  - Désigne l'ensemble des chaînes de caractères que nous pouvons former en regroupant des caractères. (ex: 'nuit', 'Couleur')

- **Opérateurs du type chaîne:**
  - Les opérateurs de relation : =, ≠
  - La fonction *longueur* qui fournit la longueur d'une chaîne.

    Exemple: longueur ('nuit') = 4.

2.6 Les types élémentaires

- **Exercice:**
  - Que sera-t-il affiché sur l'écran après l'exécution de l'algorithme suivant:

Algorithme algo

Variable A, B : entier

Début

A ← 5

B ← 9

Ecrire ('A+B')

Ecrire (A+B)

Fin

(Écran)

A+B

14

2.7 Les expressions

- Il y a deux sortes d'expressions : les expressions **arithmétiques** et les expressions **logiques**. Une expression est formée d'opérandes et d'opérateurs. Un opérande peut être une constante ou une variable. L'évaluation d'une expression repose sur les **règles de priorité** entre opérateurs (il faut noter que ces règles de priorité diffèrent selon le langage de programmation utilisé). Voici la liste des opérateurs du plus prioritaire au moins prioritaire (selon le langage Fortran):

1. Opérateurs arithmétiques: *, /
2. Opérateurs arithmétiques: +, -
3. Opérateurs de relation: $<$, $\leq$, $>$, $\geq$, $=$, $\neq$
4. L'opérateur logique: non
5. L'opérateur logique: et
6. L'opérateur logique: ou

2.7 Les expressions

- **Remarques:**
  - Pour les opérateurs de même priorité l'expression est évaluée de gauche à droite.
  - S'il y a des parenthèses on commence par évaluer les plus internes.
  - Pour déterminer le type d'une expression il faut vérifier sa syntaxe (la façon d'écriture).
- **Exemples:**
  - X + Y * (-1) + (X - A / 2)
  - A + B ou C * D